**PROGRAMLAMA DİLLERİ**

Bir programlama dilini öğrenmek, anadil öğrenmeye benzer. Çocukken, ana dilimizin sözdizimini (gramer) bilmeden, başkalarının yaptığı gibi kullanmaya başlarız; yani taklit ederiz. Ancak okula başladıktan sonra grameri adım adım öğreniriz. Programlamada da genellikle bu yöntem izlenir. Programların sözdizimlerinin (syntax -gramer) dayandığı temelleri iyi anlamasanız bile, bir çocuğun ana dilini öğrenmesi gibi, o kalıpları olduğu gibi kullanmaya çalışmak gerekiyor. Yeni başlayanların bir dili öğrenmesinin en iyi yolu budur. Taklit etmek ve bolca tekrarlamak.

**PYTHON PROGRAMLAMA DİLİ**

Python üst düzey basit sözdizimine sahip, öğrenmesi oldukça kolay, modülerliği, okunabilirliği destekleyen, platform bağımsız, interaktif ve nesne yönelimli yorumlanabilir bir script dilidir. Bu dil modüler, olağandışı durumlar, dinamik yazım, oldukça yüksek dinamik veri türleri ve sınıfları ile birlikte çalışmaktadır. Python, oldukça temiz ve sade satırlarla kombine edebilmektedir. Birçok sistem çağrısına ve kütüphanesine uygun olan birden fazla ara yüze sahiptir ve C ya da C++ ortamında geliştirilebilir. Bu dil ayrıca programlanabilir ara yüz ihtiyacı gibi uygulamalarda genişletilmiş dil olarak kullanılabilmektedir. Son olarak Python oldukça portatiftir. Bu dil birçok Unix işletim sisteminde, Mac'de ve MS-DOS, Windows, Windows NT ve OS/2 işletim sistemleriyle çalışan bilgisayarlarda çalışabilmektedir.

**Python yorumlanır:** Python çalıştırıldığı zaman kaynak kodlar bir yorumlayıcı tarafından işleme alınır. Programı çalıştırmadan önce derlemenize gerek yoktur. Bu PERL ve PHP de kullanılan web yorumlayıcılarına benzer.

**Python İnteraktiftir:** Python interaktif yani etkileşimli bir programlama dilidir.

**Python Nesneye Yöneliktir:** Python, Nesneye Dayalı bir programlama dilidir ayrıca nesneye dayalı programlamayı destekler.

**Python Yeni Başlayanlar İçin:** Python, yeni başlayanlara yönelik programcılar için mükemmel bir dildir ve basit metin işleme, WWW tarayıcılar, oyunlar gibi geniş bir uygulama yelpazesinin geliştirilmesini destekler.

**Python Öğrenmesi kolaydır:** Python'da basit bir yapı ve açıkça tanımlanmış bir sözdizimi vardır. Bu, python programlama dilinin hızla öğrenilmesini sağlar.

**Python Okunması kolaydır:** Python girintili kod yazımına zorlaması sayesinde okunabilirliği ön planda tutmuştur.

**Python Taşınabilir:** Python çok çeşitli donanım platformlarında çalışabilir ve tüm platformlarda aynı arabirime sahiptir.

**Python Genişletilebilir:** Alt düzey modülleri Python yorumlayıcıya ekleyebilirsiniz. Bu modüller, programcıların araçlarını daha verimli olması için eklemelerine veya özelleştirmelerine olanak tanır.

**Python GUI Desteklidir:** Python, Windows MFC, Macintosh ve Unix'in X Window sistemi gibi birçok sistem çağrısı, kütüphane ve pencere sistemine yaratılabilir ve taşınabilir GUI uygulamalarını destekler.

**Python Ölçeklenebilir:** Python, kabuk script yerine büyük programlar için daha iyi bir yapı ve destek sağlar.

Python, birçok farklı problem sınıflarına uygulanabilen yüksek seviyeli genel amaçlı bir programlama dilidir. Bu dil, dizi işleme, Internet protokolleri, yazılım mühendisliği ve işletim sistemlerinin ara yüzü gibi alanları kapsayan büyük bir kütüphane ile birlikte gelmektedir. Oldukça stabildir. Her yeni stabil versiyon 6 ya da 18 ay arayla yayınlanmaktadır ve bu 1991 yılından beri böyle süregelmektedir. Yüksek profilli Python projelerinin arasında Mailman posta listesi yöneticisi ve Zope uygulama sunucusu bulunmaktadır. Başta Red Hat olmak üzere bir çok Linux dağıtımı, kendi yükleyicilerinin ve sistem yöneticisi yazılımlarının bir kısmını ya da tamamını yazarken Python kullanmışlardır. Python kullanan şirketler Google, Yahoo ve Lucasfilm Ltd.’dir.

**JAVA PROGRAMLAMA DİLİ**

İlk bilgisayarlar ortaya çıktığında, ilk programcılar, bilgisayara, ancak kullandıkları makinenin makine dili ile iş yaptırabiliyorlardı. Makine dilini öğrenmek zor olmak yanında, bir makine için öğrenilen makine dili, farklı marka ve modeller için geçerli olmuyordu. Bu zorluğu aşmak için önce assembler dili ve hemen arkasından üst düzey diller ortaya çıktı. Üst düzey diller, programcılara büyük rahatlık getirdi. Çünkü programcı, işletim sistemini ve makineyi düşünmeden, kaynak programını istediği dille yazabiliyordu. Örneğin, C ile yazılan bir kaynak programı, uygun derleyici ile her işletim sisteminde ve her makinede derlenip çalıştırılabiliyordu.

Şimdi adına prosedürel diller denilen FORTRAN, COBOL, Pascal, Modula, C gibi üst düzey diller uzun yıllar programcılara başarıyla hizmet ettiler; halen de etmektedirler. Ama ortada bir problem vardı. Belirli bir işletim sistemi altında çalışan belirli tür bir makinede derlenen bir kaynak program, ancak o işletim sistemi altında çalışan o tür makinelerde koşturulabiliyordu. İşletim sistemi veya makine türü değişince; yani platform değişince, program orada koşamıyordu; yeni platforma uygun bir derleyici ile yeniden derlenmesi gerekiyordu. Buna, kısaca platform bağımlılığı diyoruz. Platform bağımlılığından kastettiğimiz şey, bir kaynak programın belirli bir işletim sistemi altında çalışan belirli bir derleyici ile derlendiği ve ancak belirli bilgisayar türlerinde koşturulabildiği gerçeğidir. Örneğin, Windows işletim sistemi altında çalışan bir PC de derlenen bir bilgisayar programını Macintosh ya da Linux işletim sistemi altında çalışan bir makinede koşturmak mümkün değildir.

Piyasaya farklı işletim sistemlerinin ve çok sayıda farklı donanımın çıkması, bilgisayar ağlarının gelişmesi, bir ağa bağlı makinelerin farklı işletim sistemleri altında çalışan farklı marka ve model bilgisayarlardan oluşuyor olması problemi büyüttü. Problemin çözümü, platformdan bağımsız çalışabilecek bir dilin yaratılmasını gerektiriyordu. Bu iş, Sun firmasının geliştirdiği Java dili ile çözüldü. Gerçi Sun firması, ortaya çıkan bu büyük problemi çözmek için değil, daha basit bir amaçla yola çıktı. Elektrikli ev aletlerinin kolay kullanılmasını sağlayacak bir dil geliştirmek istiyordu. Kısa zamanda, bu işin ancak ortak bir platformda yapılabileceğini anladı. Sun’ın programcısı James Gosling, 1995 yılında her platformda çalışan Java dilini yarattı.

Gosling basit ama harika bir buluş yaptı. Farklı işletim sistemleri ve farklı donanımlarda kurulabilen ortak bir sanal makine tasarladı. Adına JVM (Java Virtual Machine) denilen bu sanal makine ücretsiz dağıtıldı. JVM her platforma kolayca kurulabilir. Java dili ile yazılan kaynak programlar, java derleyicisi ile JVM de çalışabilen bir tür makine diline dönüştürülür. Adına java byte kod denilen bu program JVM tarafından yorumlanır ve çalıştırılır. Java byte kod bir bakıma, prosedürel dillerde yazılıp derlenmiş kaynak programların işlevini yapar. Java uygulamalarını çalıştırmak isteyen herkes, kendi makinesine JVM ‘i kurabilir. Bunun için JRE (Java Runtime Environment) adlı programı internetten indirip yüklemesi yetecektir. JRE bilgisayara bir kez yüklenir. Ondan sonra Java uygulamalarının hepsi bu makinede çalışabilir. Makinede bir java uygulaması çalışacağı zaman, JRE kendiliğinden JVM sanal makinesini oluşturur. JVM gerektiğinde çalışan bir programdır; her program gibi, işi bitince ana bellekten silinir. Dolayısıyla makineye bir zarar vermesi söz konusu değildir. Günümüzde, java uygulamaları ile karşılaşan tarayıcıların çoğu, JRE’yi otomatik olarak indirip yükleme yeteneğine sahiptir.

Java basit, modern, nesne yönelimli, tip-korumalı ve C ile C++ dillerinin iyi özeliklerini almış bir dildir. Ekstra olarak, her platformda çalışabilme yeteneğine sahiptir. Bu yetenek, onu yalnız bilgisayarlarda değil, internet uygulamalarına, cep telefonlarına, oyun makinelerine, ev aletlerine kadar geniş bir yelpazede uygulama alanı bulmasını sağladı. Bu nedenle, Java’yı hem programlama dili, hem de bir ortam olarak düşünebiliriz. Bu ortamda işletim sistemi, ağlar, internet programlama, veri tabanı ve bütün orta katman teknolojileri yer alır.

Özetle, Java basit bir dildir. Kolay öğrenilir. Genel amaçlı bir dilin bütün hünerlerine sahiptir. C, C++ dillerinin iyi özeliklerini almıştır. Kullanıcı dostu ve hızlı bir uygulama geliştirme aracıdır.

**Java basittir**: Java’yı tasarlayanlar kaynak programın kolay yazılabilmesini, kolay derlenmesini ve kolay düzeltilmesini (debug) amaçladılar. Java’nın C++ diline göre çok daha basit olmasının temel nedeni, otomatik bellek tahsisi yapması ve işi biten nesneleri bellekten yok etmesidir (garbage collection).

**Java nesne yönelimlidir**: Nesne yönelimli programlama paradigmasının bütün avantajlarını taşır. Programcıya kalıtım, polimorfizm, modular programlama, hata ayıklama (debug) ve kodların yeniden kullanılabilmesi gibi önemli yetenekleri sunar.

**Java dağıtıktır**: Bir ağ üzerindeki birden çok farklı bilgisayarın bütünleşik bir sistem olarak bir arada çalışmasını sağlar.

**Java çoklu iş yapma yeteneği vardır**. Çoklu iş yapma niteliği, bilgisayarın aynı anda birden çok işi yapabilmesi demektir. Başka dillerde sistemle ilgili prosedürlerin çağrılmasıyla yaptırılan multithreaded özeliği java dilinin özünde vardır. Multithreaded yeteneği, özellikle görsel programlamada ve ağ programlamada önem kazanır.

**Java platform bağımsızdır**: Java programları harklı platformlar için ayrı ayrı değil, JVM için bir kez derlenir. Derleme sonunda ortaya çıkan java bytecode JVM tarafından yorumlanır. Bytecode, JVM yüklü her makinede çalışabilir. JVM sanal makinesi her makineye kolayca ve ücretsiz yüklenebilir. Dolayısıyla, java programları bir kez yazılır ve her yerde çalışır.

**Java taşınabilir**: İşletim sisteminden ve donanımdan bağımsız oluşu nedeniyle, Java Bytecode bir bilgisayar sisteminden farklı bir başkasına kolayca taşınır. Aynı java programının farklı sistemlerde sorunsuz çalışabilme yeteneği, programlama alanında geniş ufuklar açmıştır.

**Java sağlamdır:** Başka dillerin ancak koşturma anında belirleyebileceği hataları, java derleme anında belirler. Güçlü hata ayıklama (debug) yeteneği vardır.

**Java güvenlidir:** Java dili, derleyicisi ve yorumlayıcısı güvenlik öncelikli olarak tasarlanmıştır. Tasarımında güvenliği öne çıkaran ilk dildir.

**R PROGRAMLAMA DİLİ**

R istatistiksel ve ekonometrik hesaplar gibi birçok özelliğin yanı sıra veri manipülasyonu, programlama ve grafiksel gösterimi bir arada sunan bütünleşmiş bir yazılım ortamıdır. R dili S ve daha sonra geliştirilen S Plus dillerine bir türevi olup ilk olarak 1991 yılında Auckland Üniversitesi İstatistik bölümünde Ross Ihaka ve Robert Gentleman tarafından geliştirilmiş ve 1993 yılında duyurulmuştur. Geliştiricilerin ikisi de isimlerinin ilk harfi olan R olması programlama diline ismini vermiştir.

R dili S diline çok benzemekte ve hemen hemen tüm işletim sistemlerinde çalışmaktadır. En basitten karmaşığa kadar çok farklı programlama seviyeleri için uygun bir ortam sunmaktadır. R’nin son yıllardaki popülerliğinin artmasının sebebi kullanım kolaylığı açık bir yazılım olması ve kullanım alanları ile başvuru kaynaklarının fazlalığıdır. Açık programlama dili olduğu için sık sık yeni sürümleri ve hata çözümleri yayınlanmaktadır. R Programı son yıllarda giderek artan bir şekilde veri bilimi dendiğinde akla gelen programlama dili olmaya başlamıştır. R ile veri analizi, veri madenciliği ya da yapay öğrenme çalışmalarının yanında diğer programlama dillerindeki gibi bilgisayar programları da yazılabilmektedir. R, Microsoft Excel de yapılabilecek basit analiz işlemlerden C, C++, Java gibi bir programlama dili ile yazılabilecek Çok karmaşık programları kadar çok çeşitli analiz ve çalışmaları izin vermektedir.

Etkin bir veri işleme ve depolama

Dizi ve matris hesaplamaları için komutlar grubu

Veri analizi için ileri düzeyli teknikler topluluğu

Verinin ekranda ya da basılı bir eserde görüntülenmesine olanak veren geniş grafiksel özellikler

Veri analizi için kullanılabilecek uyumlu ve bir arada kullanılabilen araçlar içerir.

Nesne yönelimli bir programlama dili olması

2D, 3D gelişmiş grafik araçlarına sahip olması

R ilgili en dikkat çekici gelişme Microsoft firmasının R tabanlı istatistik ve veri analitiği çözümleri sunan Revolution Analitics firmasında 23 Ocak 2015 tarihinde bünyesinde katması olmuştur. HP, Microsoft, Facebook, Google, Twitter, Foursquare, Mozilla, The New York Times, Uber, Bank of America ve daha birçok firma veri analitiği çalışmaları için programını kullanmaktadır. Programcıları C veya C++’daki gibi hızlı ve etkin kullanım değil istenen görevi yerine getirilmesini odaklanmaktadır. Bu da R ile ilgili gelişmekte olan bir konudur.

**GO Programlama Dili**

Go Programlama Dili (Golang), 2007 yılında Robert Griesemer, Rob Pike ve Ken Thompson tarafından geliştirilen, şimdilerde ise Google’ın geliştirmeye devam ettiği açık kaynaklı ve tamamen ücretsiz bir programlama dilidir. Go, Golang veya Google Go olarak adlandırılan programlama dili; C ve Algol gibi programlama dillerine benzer şekilde; derlenmiş, statik yazımlı, garbage Collection’lı bir dildir. Go programlama diline ilişkin olarak tüm diğer aracı yazılımlar ve derleyiciler de Google tarafından geliştirmektedir.

Go programlama dili geliştirilmeye başlandığı tarihten itibaren uzun bir test sürecine tabii tutuldu. Onlarca Google mühendisinin dâhil olduğu projenin ilk meyveleri 2012 yılında Go’nun kararlı sürümü olarak yayınlanan Go 1.0 ile birlikte toplanmaya başlandı. Sunucu tarafındaki avantajları ve kolayca öğrenilebiliyor olması, Go’nun kısa sürede büyük bir kitle tarafından kabul görmesine yardımcı oldu. Adobe, BBC, IBM, Intel ve Medium gibi birçok tanınmış şirket, Go programlama dilini kullanmaya başlayarak Go’nun avantajlarından faydalandı.

Go’yu diğer programlama dillerinden ayıran önemli farklılık çağın gerekliliklerine uygun ve kendiliğinden Multi-threading olarak tasarlanmış bir programlama dili olmasıdır. Google’ın programlama dili Go’yu diğer dillerden farklı kılan tüm bu donanım ve yazılım sorunlarına kendi bünyesinde çözümler üretmeyi başarması, güvenli ve performansı yüksek sistemler oluşturmaya yardımcı olmasıdır. Google’ın Go programalama dilini geliştirmesinin ardında yatan birçok önemli neden var. En önemli nedeniyse performansı yüksek ve modern bir programlama dili ortaya çıkarmaktır.

C ve C++ gibi programlama dillerinin Java ve Python gibi programlama dillerine oranla daha yüksek performanslı olmasının nedeni, bu dillerin derlenmesi sırasında yorumlayıcıların kullanılmayışından kaynaklanır. Örneği Java dilinde yazılan bir byte kodu, önce sanal makineler yardımıyla Binary koduna dönüştürülür ve ardından işlemciye ulaştırılır. Go, tıpkı C ve C++ gibi arada bir yorumlayıcı kullanılmadan doğrudan derlenir ve donanımla iletişim kurar. Fakat Go C ve C++ arasında önemli bir fark vardır. Bu dillerde yazılan kodlar doğrudan insanların okuyabileceği kodlar olamazken, Go’da yazılan kodlar insanların okuyabileceği şekilde tasarlanmıştır. Bu sayede Go, kendisiyle çalışmayı daha kolay ve verimli kılmaktadır.

Go programlama dilindeki esas, bir dilin gereksiz tüm bütünlerini ayırıp onu sadece en yararlı özelliklerle donatmaktır. Google’ın ve dolayısıyla Go programlama dilinin bu felsefesi, Go’yu öğrenmeyi katbekat kolaylaştırır. Google, Go’yu yalnızca dışarıya dönük ihtiyaçlar için hazırlamamıştır. Go’yu başta kendi sunucuları, sonrasında ise tüm internet dünyası için düşünerek geliştirmiştir. Google’da bir kodun üzerinde onlarca mühendisin çalışabilmesi için kodların anlaşılabilir olması ve yazılımcılarının yazdığı kodlarının birbiriyle uyum içinde çalışması gerekir. Go’nun basit ve kurallı bir Syntax’e sahip olmasının nedeni budur. Bir programlama dilinde istediğiniz bir şeyi meydana getirmenin birden fazla yolu olabilir. Ancak söz konusu olan Go ise sadece tek bir doğru vardır ve diğer istisnalar kabul edilemez. Böylece kod dosyası üzerinde değil onlarca, yüzlerce yazılımcı çalışsa bile birbirlerinin hazırladıkları kodlar daima uyum içinde çalışacaktır.

Go programlama dilinin diğer diller karşısındaki farklılıkları bu kadarla sınırlı değil. Go, diğerlerinin aksine “Classes” adını verdiğimiz Sınıflar yapısını kullanmıyor ve bunun yerine her şey paketlerin içine yerleştiriliyor. Sınıflar yerine Google Go’da “yapılar” ön plana çıkıyor. Modern programlama dillerinden alışkın olduğumuz inheritance desteği, constructors, açıklama metinleri (annotations) de Go’nun yapısında yer almayan diğer tuhaflıklardan ibaret.

Google Go programlama diliyle tek yapmanız gereken ihtiyacınız olanı düşlemek ve birkaç satır kod yazarak modern ve hızlı kabul edilen programlama dillerinden daha başarılı işler çıkartmak. Ardında Google’ın yer aldığı Go’nun yıldızının her geçen yıl parladığı kesin. Shopify’ın kurucu başkanı Tobias Lütke’un dediği gibi “Go, geleceğin sunucu dili olacaktır.” İddiasının gerçekçiliğiniyse ancak gelecekte görebileceğiz.

Go dilinin birincil amacı sistem programlamalarının yapılmasıdır. Aktif biçimde sunucu tarafında kullanılması için geliştirilen Go’yu, sunucu ve alt sistemler geliştirmek için harika bir dil olarak görebiliriz. Bunun yanı sıra Go ile web namına hızlı çalışan ve verimli projeler de üretilebilir. Henüz kısıtlı örneklerle karşımıza çıkıyor olmasına rağmen Google’ın, Go’yu mobil işletim sistemi Android’de kullanma planı vardır. Gelecekte Go’yu bilen geliştiriciler sadece sunucu ve sistem tarafındaki işleri halletmekle kalmayarak, kendi Android mobil uygulamalarını da Go ile verimli biçimde geliştirebilecekler.

Go gömülü sistemlerde kullanılan bir programlama dili olarak da karşımıza çıkabilir. Geliştirici topluluğunun büyük bir bölümü buna inanıyor olsa bile, gömülü sistemlerde hali hazırda süregelen C ve C++ haçlı savaşları nedeniyle bunu şimdilik kestirmek tam olarak mümkün olmayabilir.

Go ile neler yapabileceğiniz kadar neler yapamayacağınızı bilmek de önemli. Go’yu bir betik dili olarak kafanızda resmetmelisiniz. Go’nun geliştirilmesinin amacı her ne kadar daha modern olsa da Java veya Python’u tahtından indirip, kendini tahta çıkarmak değildir. Go daha ziyade C ve Python gibi dillerle birlikte çalışarak başarılı, güçlü, modern ve verimli sistemler oluşturmak amacıyla geliştirilmiştir.

|  |  |  |  |
| --- | --- | --- | --- |
| **Python** | **Java** | **Go** | **R** |
| Genel amaçlı programlama dilidir | Üst düzey bir programlama dilidir | Yüksek Performanslı bir programlama dilidir | İleri düzey veri analizi için kullanılan bir dildir |
| Platform bağımsızdır | Platform bağımlıdır | Platform bağımsızdır | Platform bağımsızdır |
| Statik yazılımlıdır | Dinamik yazılımlıdır | Statik yazılımlıdır | Dinamik yazılımlıdır |
| Hızlıdır | Yavaştır | Hızlıdır | Yavaştır |
| Nesneye tabanlı programlama dilidir | Nesneye tabanlı programlama dilidir | Nesneye tabanlı programlama dilidir | Nesneye tabanlı programlama dilidir |
| İnteraktiftir | Dağıtıktır |  |  |
| Öğrenmesi kolaydır | Öğrenmesi kolaydır | Öğrenmesi kolaydır | Öğrenmesi zordur |
|  |  |  |  |